N ASA-CR-172,253

NASA-CR-172253
19840005312

NASA Contractor Report 172253

ICASE

AN APPLICATION OF NONLINEAR PROGRAMMING TO THE DESIGN OF REGULATORS USING A LINEAR-QUADRATIC FORMULATION

Peter Fleming

Contract No. NAS1-17070
October 1983

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

THE APPLICATION OF NONLINEAR PROGRAMMING TO THE DESIGN OF REGULATORS USING A LINEAR-QUADRATIC FORMULATION

Peter Fleming
Institute for Computer Applications in Science and Engineering
and
University College of North Wales
United Kingdom

ABSTRACT

A design technique is proposed for linear regulators in which a feedback controller of fixed structure is chosen to minimize an integral quadratic objective function subject to the satisfaction of integral quadratic constraint functions. Application of a nonlinear programming algorithm to this mathematically tractable formulation results in an efficient and useful computer-aided design tool. Particular attention is paid to computational efficiency and various recommendations are made. Two design examples illustrate the flexibility of the approach and highlight the special insight afforded to the designer.

Research supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-17070 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, VA 23665.

Introduction

Research into control system optimization has followed a number of separate paths following the interest generated by the linear quadratic regulator (LQR) approach. Although it possesses attractive properties the LQR method suffers from two main drawbacks:

i) the resulting controller requires access to the full set of plant states or to a state reconstructor, and,

ii) the scalar quadratic cost function is inadequate for the representation of a set of design objectives.

Levine and Athans (1970) pioneered the suboptimal regulator (SOR) approach by prespecifying the feedback controller structure but retaining the quadratic cost function adopted in the LQR design. Various features can be incorporated within the SOR context by expanding the cost function to include such measures as trajectory sensitivity and model-following (see Fleming 1979). Computationally this is an attractive approach but it still suffers from the inadequacy of the scalar quadratic measure to describe all the different facets of system performance.

Another line of attack was prompted by Schy, Adams and Johnson (1973) and Zakian (1973) in which system constraints and specifications are represented by a set of simultaneous algebraic inequalities. Thus the problem description could incude such basic control design parameters as overshoot, damping, settling time, etc. Polak and Mayne advanced this approach by posing semi-infinite programming problems in which design objectives are realized by minimizing a function subject to a set of inequalities where these objectives can be expressed as infinite dimensional constraints (Mayne, Polak and Sangiovanni-Vincentelli 1982). Such problems require special mathematical programming algorithms to handle the infinite dimensional constraints and, sometimes, nondifferentiable functions.

The linear quadratic constrained regulator (LQCR) method described here strikes a compromise between these two approaches. Common to all is the prespecification of controller structure; here an integral quadratic objective function is minimized subject to a set of integral quadratic constraint functions which may represent bounds on control energy, sensitivity measures, model-following errors, etc. An efficient solution procedure, based on readily available software, arises from this formulation and has led to an effective computer-aided design package being constructed on a 32-bit minicomputer.

Two design examples which illustrate some novel features of the method are presented. Various controller configurations are studied for helicopter regulation where it is found that the identification of active constraints affords the designer additional insight into the problem. In a flight control example sensitivity reduction is an objective and a trade-off curve proves useful in selection of a suitable controller.

2. A Typical LQCR Problem

Given a linear time-variant plant

$$\dot{\underline{x}}_p = A_p\underline{x}_p + B_p\underline{u}_p, \qquad \underline{y}_p = C_p\underline{x}_p,$$

where

$$\underline{x}_p = n \times 1 \quad \text{plant vector}$$

$$\underline{u}_p = m \times 1 \quad \text{control vector}$$

and

$$\underline{y}_p = r \times 1 \quad \text{output vector,}$$

and a fixed controller structure,

$$\underline{u}_p = K_o\underline{y}_p,$$

a typical LQCR design might seek a controller gain matrix, K_0, to regulate the output responses subject to certain control energy limitations, i.e.

$$\underset{\text{w.r.t.}K_0}{\text{minimize}} \int_0^\infty \underline{y}_p^T Q_0 \underline{y}_p \, dt,$$

subject to

$$\int_0^\infty u_{p_\ell}^2 \, dt \leq z_\ell, \qquad \ell=1,2,\cdots,m,$$

where z_ℓ are constraint bounds.

In a corresponding SOR design this problem would be approximately solved by successive minimizations of

$$J = \int_0^\infty \{\underline{y}_p^T Q_p \underline{y}_p + \underline{u}_p^T R_p \underline{u}_p\} dt,$$

where the designer strives to find the appropriate choice of Q_p and R_p to satisfy the control energy constraints.

Thus what was previously solved in a number of unconstrained optimization designs (SOR) is now accomplished in a single constrained optimization design (LQCR). It is this "one-pass" solution procedure which makes the approach so appealing to a designer. Although an LQCR solution requires more computing time than an SOR solution it requires considerably less time than the sequence of SOR solutions necessary to solve the problem. Moreover use of the linear-quadratic formulation leads to suprisingly modest computing time overheads when compared with an SOR solution.

3. Design Procedure

The design options available in the LQCR program are closely related to those found to be useful in an earlier SOR program (Fleming 1979). The design procedure summarized here is simply intended to be representative of the underlying concept. Future users will invent new design options appropriate to their needs.

The linear time-invariant plant description is

$$\dot{\underline{x}}_p = A_p \underline{x}_p + B_p \underline{u}_p, \qquad \underline{x}_p(0) = \underline{x}_{p_0} \tag{1}$$

$$\underline{y}_p = C_p \underline{x}_p, \tag{2}$$

and the controller configuration options available to the designer are:

i) Full state feedback

$$\underline{u}_p = K_s \underline{x}_p \tag{3}$$

ii) Output feedback

$$\underline{u}_p = K_0 \underline{y}_p \tag{4}$$

iii) Dynamic compensation

$$\underline{u}_p = A_c \underline{y}_p + B_c \underline{x}_c, \tag{5}$$

where $\underline{x}_c$ is an $s \times 1$ compensator vector which satisfies the dynamic equation

$$\dot{\underline{x}}_c = C_c \underline{y}_p + D_c \underline{x}_c, \qquad \underline{x}_c(0) = 0. \tag{6}$$

Having selected a controller configuration the designer specifies the objective and constraint functions for the nonlinear programming problem:

$$\text{minimize} \quad J_0$$

such that

$$J_\ell \leqslant z_\ell, \qquad \ell=1,2,\cdots,q,$$

where z_ℓ are constraint bounds and the set of cost functions, J_ℓ, $\ell=0,1,\cdots,q$ have the basic infinite-time quadratic integral form:

$$J_\ell = \int_0^\infty \left(\underline{x}_p^T Q_{p_\ell} \underline{x}_p + \underline{u}_p^T R_{p_\ell} \underline{u}_p\right) dt, \qquad \ell=0,1,\cdots,q.$$

These cost functions may be expanded or modified according to the circumstances of the design. For example, a quadratic penalty term, $\dot{x}_c R_{c_\ell} \dot{\underline{x}}_c$, may be included if the dynamic compensation option is selected.

For the case where A_p and B_p (equation (1)) are functions of a scalar time-invariant parameter, α, a differential trajectory sensitivity vector, $\underline{x}_s = \partial \underline{x}_p / \partial \alpha$, is introduced which satisfies the equation

$$\dot{\underline{x}}_s = A_s \underline{x}_p + B_s \underline{u}_p + A_p \underline{x}_s + B_p \partial \underline{u}_p / \partial \alpha, \qquad \underline{x}_s(0) = 0,$$

derived by partially differentiating equation (1) with respect to α, where $A_s = \partial A_p / \partial \alpha$ and $B_s = \partial B_p / \partial \alpha$. If sensitivity reduction is an objective then the quadratic sensitivity measure, $\underline{x}_s^T Q_{s_\ell} \underline{x}_s$, may be included in the cost functions.

In order to monitor how well a set of plant trajectories match a set of "model" trajectories either virtual model-following (VMF) or implicit model-following (IMF) terms may be implemented. A quadratic measure of the difference between plant and model <u>outputs</u>, $(\underline{y}_p - \underline{y}_m)^T Q_{p_\ell} (\underline{y}_p - \underline{y}_m)$, (VMF) or <u>state derivatives</u>, $(\dot{\underline{x}}_p - \dot{\underline{x}}_m)^T Q_{p_\ell} (\dot{\underline{x}}_p - \dot{\underline{x}}_m)$, (IMF) replaces the usual state measure, where the model response is defined by

$$\dot{\underline{x}}_m = A_m \underline{x}_m,$$

$$\underline{y}_m = C_m \underline{x}_m,$$

$\underline{x}_m = v \times 1$ model state vector and $\underline{y}_m = r \times 1$ model output vector. Weighting matrices Q_{p_ℓ}, Q_{s_ℓ}, R_{p_ℓ} and R_{c_ℓ} will usually be diagonal and while it is possible to include more than one cost function option in either the objective or constraint functions it will be more beneficial in an LQCR design to identify these features separately. Indeed a cost function need only focus on a single vector element thus allowing the designer to exercise very precise control over competing functions in the minimization process.

4. Formulation of Nonlinear Programming Problem

Depending on the selected design options the plant state vector, $\underline{x}_p$, will be augmented to incorporate a compensator state vector, $\underline{x}_c$, a trajectory sensitivity vector, $\underline{x}_s$, and a model state vector, $\underline{x}_m$ to form the system state vector, $\underline{x}$. This leads to the following concise expressions for the system state equation and cost functions:

System $$\dot{\underline{x}} = \bar{A}(K)\underline{x}, \qquad \underline{x}(0) = \underline{x}_0, \tag{7}$$

where $$\bar{A}(K) = A + B_1KC_1 + B_2KC_2 \tag{8}$$

Objective/Constraint Functions

$$J_\ell = \int_0^\infty \underline{x}^T\bar{Q}_\ell(K)\underline{x}dt, \qquad \ell=0,1,\cdots,q, \tag{9}$$

where

$$\overline{Q}_\ell(K) = Q_\ell + C_1^T K^T R_\ell K C,. \tag{10}$$

Matrices $A, B_1, B_2, C_1, C_2, Q_\ell$ and R_ℓ are easily derived from the input matrices $A_p, A_m, A_s, B_p, B_s, C_p, C_m, Q_{p_\ell}, Q_{s_\ell}, R_{c_\ell}$ and R_{p_ℓ}; their composition is described in Fleming (1979). Matrices $\overline{A}$ and $\overline{Q}_\ell$ are functions of the gain matrix, K, containing the optimization parameters, and it may have one of three constructions:

i) Full state feedback

$$K = K_s$$

ii) Output feedback

$$K = K_0$$

iii) Dynamic compensation

$$K = \begin{bmatrix} A_c & B_c \\ C_c & D_c \end{bmatrix}.$$

Note that it is not imperative for all of the elements, k_{ij}, of K to be variable: some may be fixed to zero or constant values to aid investigation of gain redundancy, specific compensator structures, etc. (See Fleming 1981). Collecting the variable elements of K into a parameter vector, $\underline{k}$, we have the nonlinear programming problem:

$$\underset{\text{w.r.t. } \underline{k}}{\text{minimize}} \; J_0 \tag{11}$$

subject to the inequality constraints

$$J_\ell - z_\ell \leq 0, \qquad \ell=1,2,\cdots,q, \tag{12}$$

where the cost functions are governed by the system state equation (7).

Given a particular value for K the cost functions, $J_\ell, \ell = 0,1,\cdots q$, (9), can be computed from

$$J_\ell = \mathrm{tr}\left(P_\ell X_0\right), \qquad \ell=0,1,\cdots,q, \tag{13}$$

where $X_0 = \underline{x}_0\underline{x}_0^T$ and P_ℓ satisfies the Liapunov matrix equation

$$P_\ell\overline{A} + \overline{A}^T P_\ell = -\overline{Q}_\ell, \qquad \ell=0,1,\cdots,q, \tag{14}$$

provided that $\overline{A}$ is a stable matrix.

Analytic expressions for the gradients of the cost functions (13) with respect to the controller gain matrix K, are derived here using an approach similar to that of Wilson (1970). Differentiating (13) with respect to a gain element, k_{ij}, of matrix, K, we have

$$\frac{\partial J_\ell}{\partial k_{ij}} = 2\mathrm{tr}\left(\Lambda P_\ell \frac{\partial \overline{A}}{\partial k_{ij}}\right) + \mathrm{tr}\left(\Lambda \frac{\partial \overline{Q}_\ell}{\partial k_{ij}}\right), \tag{15}$$

where Λ satisfies the Liapunov matrix equation

$$\Lambda\overline{A}^T + \overline{A}\Lambda + X_0 = 0, \tag{16}$$

and combining (8), (10) and (15) it follows that

$$\frac{\partial J_\ell}{\partial K} = 2\left(B_1^T P_\ell \Lambda C_1^T + B_2^T P_\ell \Lambda C_2^T + R_\ell K C_1 \Lambda C_1^T\right). \tag{17}$$

The gradient vector, $\partial J_\ell/\partial \underline{k}$, can be easily constructed from (17).

It is evident from (13) that the solution is initial condition dependent, however when initial conditions are unknown, matrix X_0 may be modified so that either $E\{J_\ell\}$, $\underset{\underline{x}_{p_0}}{\text{ave}}\, J_\ell$ or $\underset{\underline{x}_{p_0}}{\max}\, J_\ell$ are evaluated (see Fleming 1979).

It is a simple matter to extend the nonlinear programming problem, (11) and (12), to include direct constraints on gain parameters. Expressed as a general algebraic expression

$$f_\ell(\underline{k}) \leqslant 0, \qquad \ell=1,2,\cdots,w \tag{18}$$

these constraints may simply represent bounds on certain variable gains, k_{ij}. Analytic gradients are obtained in a straightforward manner.

5. Solution Procedure

5.1. Nonlinear Programming Algorithm

Following a short survey of algorithms using the ADS program package (Vanderplaats 1983) it was established that the use of analytic gradients (17) is preferred to that of using finite difference approximations, resulting in improved accuracy and solution times. Four algorithms were compared:

a) Method of feasible directions (Zoutendijk 1960; Vanderplaats 1973),

b) Sequential unconstrained minimization technique (SUMT) using the quadratic exterior penalty function method (Fiacco and McCormick 1968),

c) SUMT using the quadratic extended interior penalty function method (Haftka and Starnes 1976), and

d) Augmented Lagrangian multiplier method (Powell 1978).

Although (c) was the most accurate, (a) was found to be the most satisfactory algorithm within the CAD context, yielding solution times 6-8 times faster than (b), (c) and (d) for comparable accuracy levels. The promising iterative quadratic programming method of Powell (1978) will be tested using a forthcoming version of ADS.

Zoutendijk's Method of Feasible Directions initially finds a feasible point and then proceeds by iteratively searching along feasible directions. In the program package the BFGS variable metric method (e.g. Fletcher 1970) is employed within the feasible directions method if no constraints are violated. At each iteration gradient information is required only for the active or violated constraints. The objective and constraint functions are computed at each iteration and within the line search procedures for each new estimate of K.

5.2. Liapunov Matrix Equations (LMEs)

A breakdown of CPU usage reveals that function and gradient evaluations dominate the solution time and these evaluations, in turn, are subject to the efficiency of the solution of the LMEs, (14) and (16). Generally recognized as the most efficient general LME solver, the method of Bartels and Stewart (1972) possesses additional properties which may be exploited in this application. It is a transformation method which reduces $\bar{A}$, (14), to its real Schur form and then obtains a solution by solving sets of linear systems whose individual orders do not exceed four. Therefore only one Schur reduction of $\bar{A}$ per estimate of K is required for the solution of the $(q+2)$ LMEs, (14) and (16), when computing the objective and constraint functions, (13), and their gradients, (17). The solution of the LMEs is

further accelerated by exploiting certain structural and sparseness properties of $\bar{A}, \bar{Q}_\ell$ and X_0. The computing time for the Bartels-Stewart algorithm is proportional to N^3, where N is the order of the LME.

It should be noted that there are circumstances under which the LMEs are more efficiently solved using the direct method of MacFarlane (1963). Although the computing time for this method for one LME solution is proportional to N^6, subsequent solutions for different $\bar{Q}_\ell$ (14) rely only on the application of back substitution. However the execution time superiority of this method only prevails for certain combinations of small N and large q (12). For large N the Bartels-Stewart algorithm is the most efficient and is the recommended technique.

5.3. Obtaining and Maintaining a Stable Matrix $\bar{A}$

While it is permissible to start the optimization from an infeasible point with respect to the constraints (12) and (18), the solution of (14) demands that the initial parameter vector, k, and subsequent estimates of this vector stabilize the system (7). An expression constraining eigenvalues of $\bar{A}$ to have negative real parts cannot be included in the nonlinear programming problem description since violation of this constraint invalidates the solution of (14). A steepest descent technique similar to that of Koenigsberg and Frederick (1970) is recommended to search for a stabilizing value of k should one be unavailable.

Subsequently the eigenvalues of $\bar{A}$ are monitored throughout the optimization search procedure. Although the problem is such that the routine will tend to generate stabilizing values of $\underline{k}$, computational traps have been set to inhibit excursions into the unstable region. Should such a violation

occur the line search step is repeatedly halved until a stable value of $\underline{k}$ is reached. Use of the Bartels-Stewart method for solving LMEs permits economic evaluation of the eigenvalues from the diagonal and principal subdiagonal elements of the Schur form of $\overline{A}$.

6. Application Examples

6.1 Alternative Controller Configurations

Helicopter longitudinal dynamics are described in Michael and Farrar (1973) for a plant which has four states, $\underline{x}_p^T = [\mu_x, \mu_z, \theta, \dot{\theta}]^T$, and two controls, $\underline{u}_p^T = [u_1, u_2]^T$. The goal is to satisfactorily regulate μ_x and μ_z (forward and vertical velocities) employing feedback from only two states, θ and $\dot{\theta}$ (pitch angle and pitch rate), since the use of airspeed sensors for μ_x and μ_z is undesirable. An LQR design, incorporating full state feedback, defines a satisfactory model response:

$$\dot{\underline{x}}_m = (A_p + B_p K_s^*)\underline{x}_m = A_m \underline{x}_m,$$

where

$$\underline{u}_p = K_s^* \underline{x}_p.$$

The aim here is to attempt to match this model response without recourse to the use of airspeed sensors and recognizing control magnitude limitations.

Thus we have the LQCR design problem in which we seek a controller having an acceptable configuration to minimize the model-following error term

$$J_0 = \int_0^\infty (\underline{x}_p - \underline{x}_m)^T Q_p (\underline{x}_p - \underline{x}_m)\,dt,$$

where Q_p = diag (1,1,0,0), subject to the control energy constraints

$$J_1 = \int_0^\infty u_1^2 dt - 4.06 \leq 0,$$

$$J_2 = \int_0^\infty u_2^2 dt - 6.57 \leq 0,$$

where the control energy bounds are those limiting the LQR design. Tabulated design results for three controller configuration besides the LQR controller are presented.

Table 1. Design results for various controller configurations

Controller	Gain Matrix, K	J_0	Active Constraints
LQR	$\underline{u}_p = \begin{bmatrix} -2.56 & 0.29 & 0.50 & 0.24 \\ 0.38 & 1.98 & -0.26 & -0.06 \end{bmatrix} \underline{x}_p$	0	J_1, J_2
1	$\underline{u}_p = \begin{bmatrix} 0.16 & 0.14 \\ -0.25 & 0.22 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$	0.072	J_2
2	$\underline{u}_p = \begin{bmatrix} 0.18 \\ -0.25 \end{bmatrix} \theta$	0.35	J_2
3	$\begin{bmatrix} \underline{u}_p \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} 0.32 & -0.24 \\ -0.06 & -0.30 \\ 1 & -1.37 \end{bmatrix} \begin{bmatrix} \theta \\ x_c \end{bmatrix}$	0.10	—

Identification of the active constraint for the design of Controllers 1 and 2 indicates that relaxation of the control energy bound on u_2 will lead to improved model-following. It is clear from Figs. 1 and 2 that excellent model-following is achieved by Controller 1, while Controller 2, employing pitch angle feedback only, obtains less satisfactory results.

Implementing controller dynamics with pitch angle feedback (Controller 3) produces model-following results comparable to those of Controller 1. For this design gain element k_{31} was fixed at unity in order to realize a minimal parameter form for the compensator.

It is interesting to find that for the control configurations 1 and 2, only control u_2 is exercised to the limit (its energy term is active at the minimum) in pursuit of the model-following objective, thus implying that increased u_1 control energy degrades the model-following capability. This is simply caused by implementation of these new control configurations and is not due to any properties of the model since the same effect was noted for an objective function containing plant output terms alone.

A fuller discussion of this example is contained in Fleming (1983), where it is shown that the control energy bounds are effective in limiting maximum control magnitudes.

6.2 Sensitivity Reduction

This example exercises two cost function options: sensitivity reduction and implicit model-following. IMF is employed in favor of VMF since it results in a lower-order system description and is sufficiently accurate for this application.

We consider the flight dynamics of an aerodynamically unstable aircraft (Grubel and Kreisselmeier 1974):

$$\dot{\underline{x}}_p = A_p(\alpha)\underline{x}_p + B_p(\alpha)u,$$

where the components of $\underline{x}_p$ are incremental longitudinal velocity, flight path angle, γ, pitch rate and pitch angle. The control input is the elevator deflection. Plant matrices A_p and B_p are dependent on a parameter α $(0.3<\alpha<0.7)$ which is the relative position of the c.g. of the aircraft. Plant and sensitivity matrices A_p, B_p, A_s, and B_s are evaluated for $\alpha_0 = 0.5$ and may be determined from Grubel and Kreisselmeier (1974). In their paper, the authors solved the LQR problem and observed the flight path angle response to the initial condition $\underline{x}_{p_0} = [0\ 1\ 0\ 1]^T$ (see Fig. 3). The nominal response $(\alpha_0 = 0.5)$ satisfies the design goal of following a step input with essentially no overshoot in a 5 percent settling time of 2s. However the trajectories for off-nominal parameter values are unsatisfactory: a sluggish response for $\alpha = 0.3$ and 73 percent overshoot for $\alpha = 0.7$.

The aim of the LQCR design is to find a controller of similar magnitude which gives a comparable flight path angle response for $\alpha = 0.5$ and is less sensitive to parameter variations. We therefore seek the full state feedback controller, $\underline{u}_p = K\underline{x}_p$, which minimizes the sensitivity measure

$$J_0 = \int_0^\infty \underline{x}_s^T Q_s \underline{x}_s dt \tag{19}$$

subject to the constraints on model-following errors,

$$J_1 = \int_0^\infty (\dot{\underline{x}}_p - \dot{\underline{x}}_m) Q_p (\dot{\underline{x}}_p - \dot{\underline{x}}_m) dt - z_1 \leq 0 \tag{20}$$

and control energy,

$$J_2 = \int_0^\infty R_p u_p^2 dt - z_2 \leq 0, \tag{21}$$

where Q_s = diag (0,1,0,0), $Q_p = I, R_p = 100$, $z_2 = 7.77$ and the model state vector is derived from the LQR response for $\alpha = 0.5$. The bound z_2 corresponds to the LQR control energy measure and the model-following error bound, z_1, is open to experimentation by the designer. It was found that the control energy constraint was ineffective in limiting the magnitude of $u_p(0)$ which played an important role in sensitivity reduction. To maintain a consistent comparison with the LQR design the following two algebraic constraints of the form (18) were added:

$$k_{12} + k_{14} - u_{p_0} \leq 0 \tag{22}$$

$$-(k_{12}+k_{14}) - u_{p_0} \leq 0, \tag{23}$$

where

$$\begin{aligned} u_p(0) &= K\,\underline{x}_{p_0} \\ &= \left[k_{11} k_{12} k_{13} k_{14}\right] \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \\ &= k_{12} + k_{14}, \end{aligned}$$

and for the LQR design, $u_p(0) = u_{p_0} = 0.60$.

LQCR designs were carried out for a range of values of z_1 to examine the expected trade-off between sensitivity reduction and model-following (see Fig. 4). This trade-off curve suggests that Controller A is a candidate for "best" design and the corresponding flight path angle response is illustrated in Fig. 5(a). Here trajectory dispersion of γ is reduced with overshoot for $\alpha = 0.7$ less than 35% while the nominal response has no overshoot and a 5 percent settling time of 3.0s. Controller B improves on model-following

capability at the expense of sensitivity reduction (see Fig. 5(b)). In both designs the active constraints are equations (20) and (22).

At this level of model-following error $(z_1=1)$ we investigate the effect of abandoning the control constraints altogether, i.e. dropping constraints (21)-(23), to find the amount of sensitivity reduction possible under these relaxed conditions. The resulting flight path angle γ, response due to Controller C is illustrated in Fig. 5(c), where we observe, in particular, that overshoot for the $\alpha = 0.7$ case is reduced to 44%; the actual sensitivity measure (19), J_0, is 3.40. Since the additional control effort,

$$\int_0^\infty 100u_p^2 dt = 8.09,$$

$$u_p(0) = 0.76,$$

is relatively minor the implication is that relaxing control constraints pays substantial dividends for this sensitivity-accented problem.

6.3. Computing Details

These examples were worked on a VAX 11/95 minicomputer which operated a conversational-mode CAD package employing the algorithms recommended in Section 5. Both examples required augmented system descriptions (7) of order 8 (order 9 for the dynamic compensation case in 6.1) and the number of minimizing variables ranged from 2 to 5. Computing times for individual designs varied according to solution accuracy demands and initial controller gain estimates but took typically 20-100 seconds of CPU time. Tests carried out on corresponding SOR designs indicated that LQCR designs required 4-8

times more computing time. It must be remembered, however, that the SOR design process is essentially iterative requiring a large number of designs before approaching the results of a single LQCR design.

7. Concluding Remarks

Nonlinear programming has been applied to regulator design in a variant of the LQR design procedure. The approach has significant advantages allowing different controller configurations to be tested and invoking sensitivity and model-following terms together with the more usual state and control terms in the design functions. It achieves design goals in a more direct and convenient manner than its SOR design counterparts. Used as a one-pass solution procedure (Section 6.1) or in the generation of trade-off curves (Section 6.2) each design yields a variety of information from inspection of objective and constraint values at the solution.

Its format is flexible, based on a linear-quadratic formulation, and may be easily modified for different user's requirements, e.g. inclusion of disturbance measures, new sensitivity approaches. The integral quadratic measures which represent the objective and constraint functions may be readily interpreted as RMS values in designs which include a noise term in the system description.

Acknowledgement

The author is grateful to Nancy Shoemaker (ICASE) for computing assistance and to A. A. Schy and D. P. Giesy (both of NASA Langley) for useful discussions in connection with this work.

References

BARTELS, R. H. and STEWART, G. W., 1972, Commun. ACM, 15, 820.

FIACCO, A. V. and McCORMICK, G. P., 1968, Nonlinear Programming: Sequential Unconstrained Minimization Techniques (Wiley).

FLEMING, P. J., 1979, Proc. IFAC Symposium on Computer-Aided Design of Control Systems (Zurich, Switzerland), Pergamon Press, 259.

FLEMING, P. J., 1981, Proc. IASTED Symposium on Modelling, Identification and Control (Davos, Switzerland), 136.

FLEMING, P. J., 1983, Proc. IFAC Workshop on Application of Nonlinear Programming to Optimization and Control (San Francisco), Pergamon Press, to be published.

GRUBEL, G. and KREISSELMEIER, G., 1974, Proc. Joint Automatic Control Conference, (Austin, Texas) 328.

HAFTKA, R. T. and STARNES, J. H., JR., 1976, AIAA J., 14, 718.

KOENIGSBERG, W. D. and FREDERICK, D. K., 1970, Proc. Joint Automatic Control Conference, (Atlanta, Georgia) 674.

LEVINE, W. S. and ATHANS, M., 1970, IEEE Trans. Automat. Control, 15, 44.

MACFARLANE, A. G. J., 1963, Quart. J. Appl. Math., XVI, 262.

MAYNE, D. Q, POLAK, E. and SANGIOVANNI-VINCENTELLI, A., 1982, Automatica, 19, 147

MICHAEL, G. J. and FARRAR, F. A., 1973, Internat. J. Control, 17, 1325.

POWELL, M. J. D., 1978, Math. Programming, 14, 224.

SCHY, A. A., ADAMS, W. M., JR., and JOHNSON, K. G., 1973, Proc. 17th AGARD Meeting on Advances in Control Systems (Geilo, Norway).

VANDERPLAATS, G. N., 1983, ADS -- A Fortran program for automated design synthesis -- Version 0.0, (Naval Postgraduate School, Monterey, California).

VANDERPLAATS, G. N., 1973, NASA Report TM X-62282.

WILSON, D. A., 1970, Proc. Inst. Elect. Eng., 117, 1161.

ZAKIAN, V., 1973, Proc. Inst. Elect. Eng., 120, 1421

ZOUTENDIJK, G. G., 1960, Methods of Feasible Directions (Elsevier, Amsterdam).

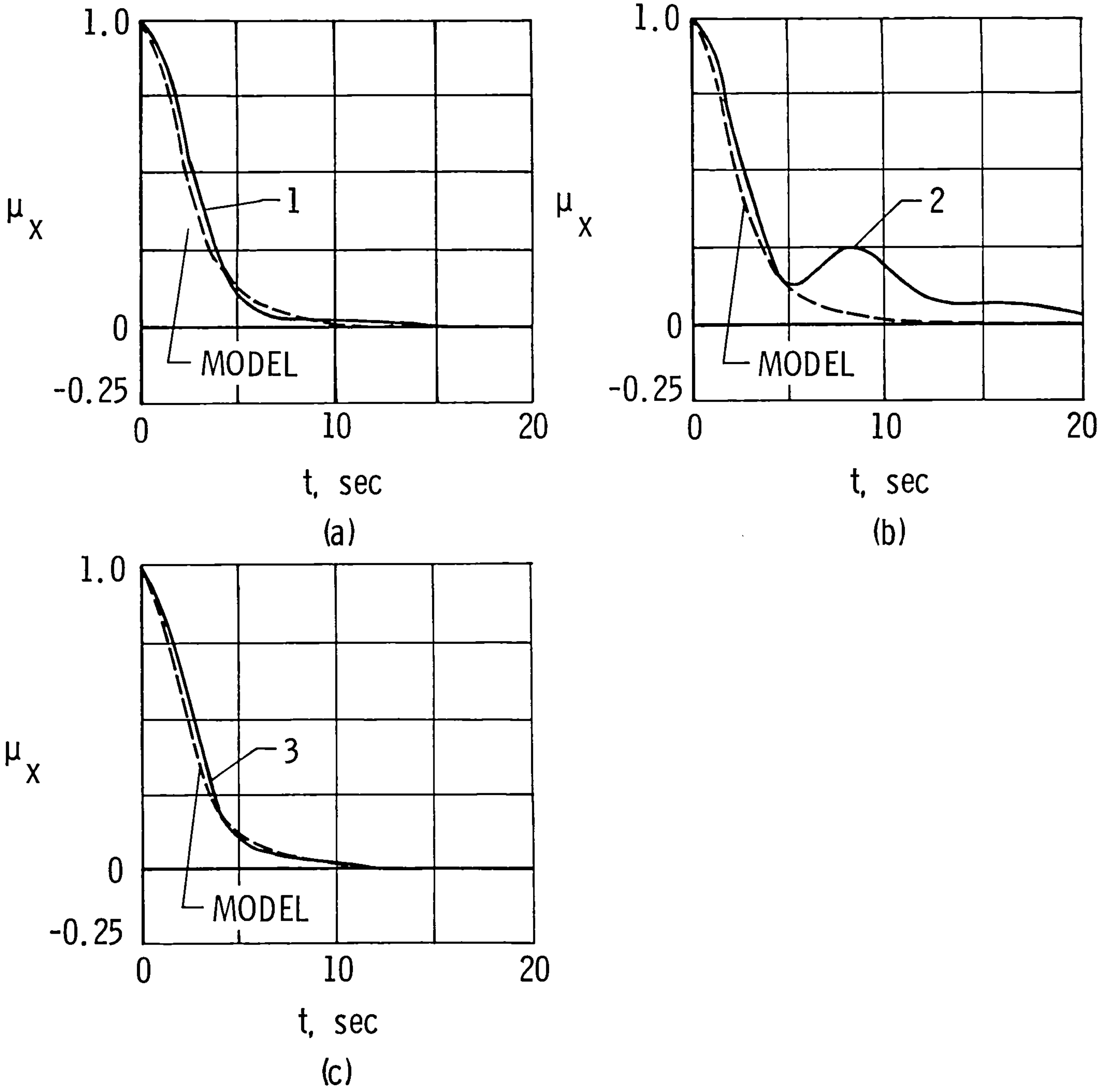

Figure 1. Forward velocity, μ_x, responses illustrating model-following capabilities of a) Controller 1, b) Controller 2 and c) Controller 3.

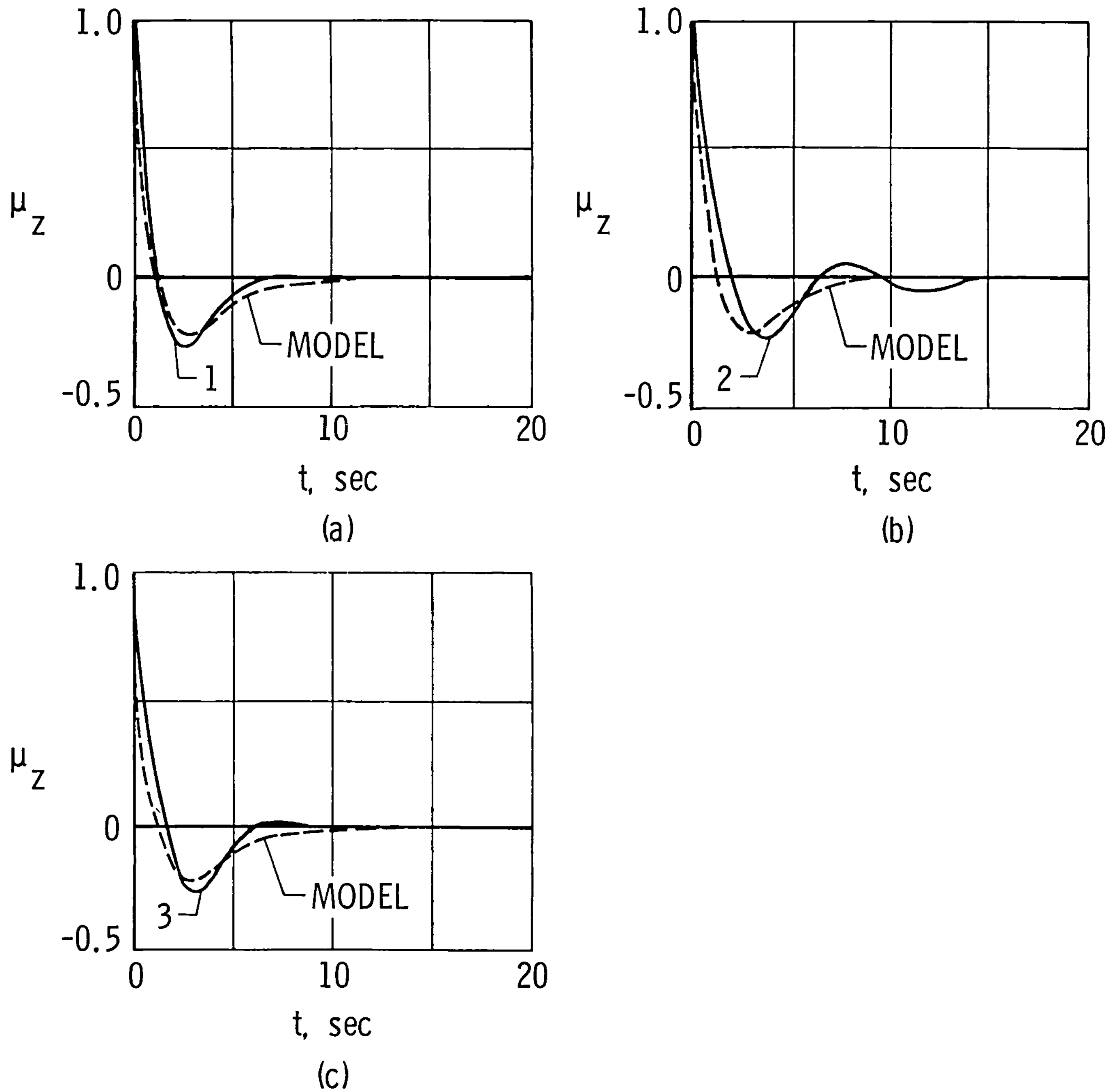

Figure 2. Vertical velocity, μ_z, responses illustrating model-following capabilities of a) Controller 1, b) Controller 2 and c) Controller 3.

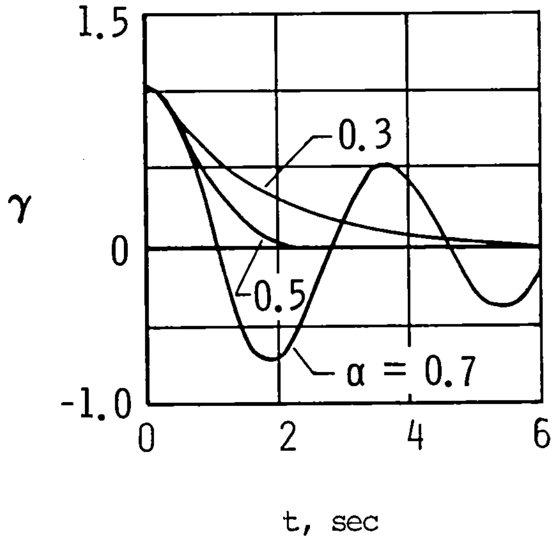

Figure 3. Flight path angle, γ, response due to LQR controller.

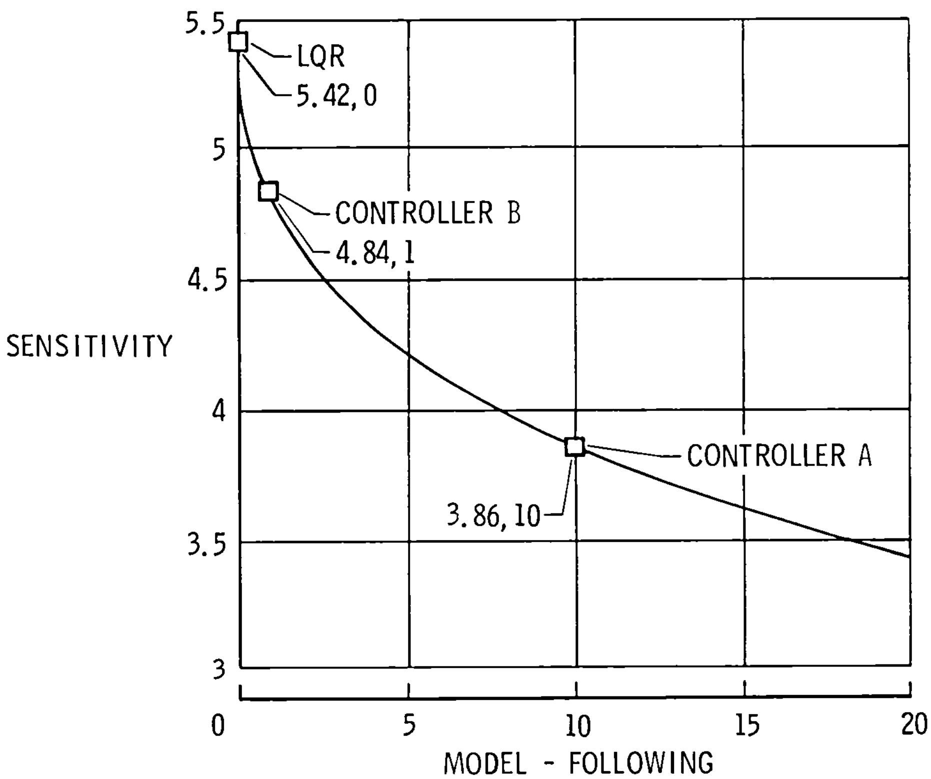

Figure 4. Trade-off curve of sensitivity, $\int_0^\infty \underline{x}_s^T Q_s \underline{x}_s dt$, v. model-following, $\int_0^\infty (\dot{\underline{x}}_p - \dot{\underline{x}}_m)^T Q_p (\dot{\underline{x}}_p - \dot{\underline{x}}_m) dt$.

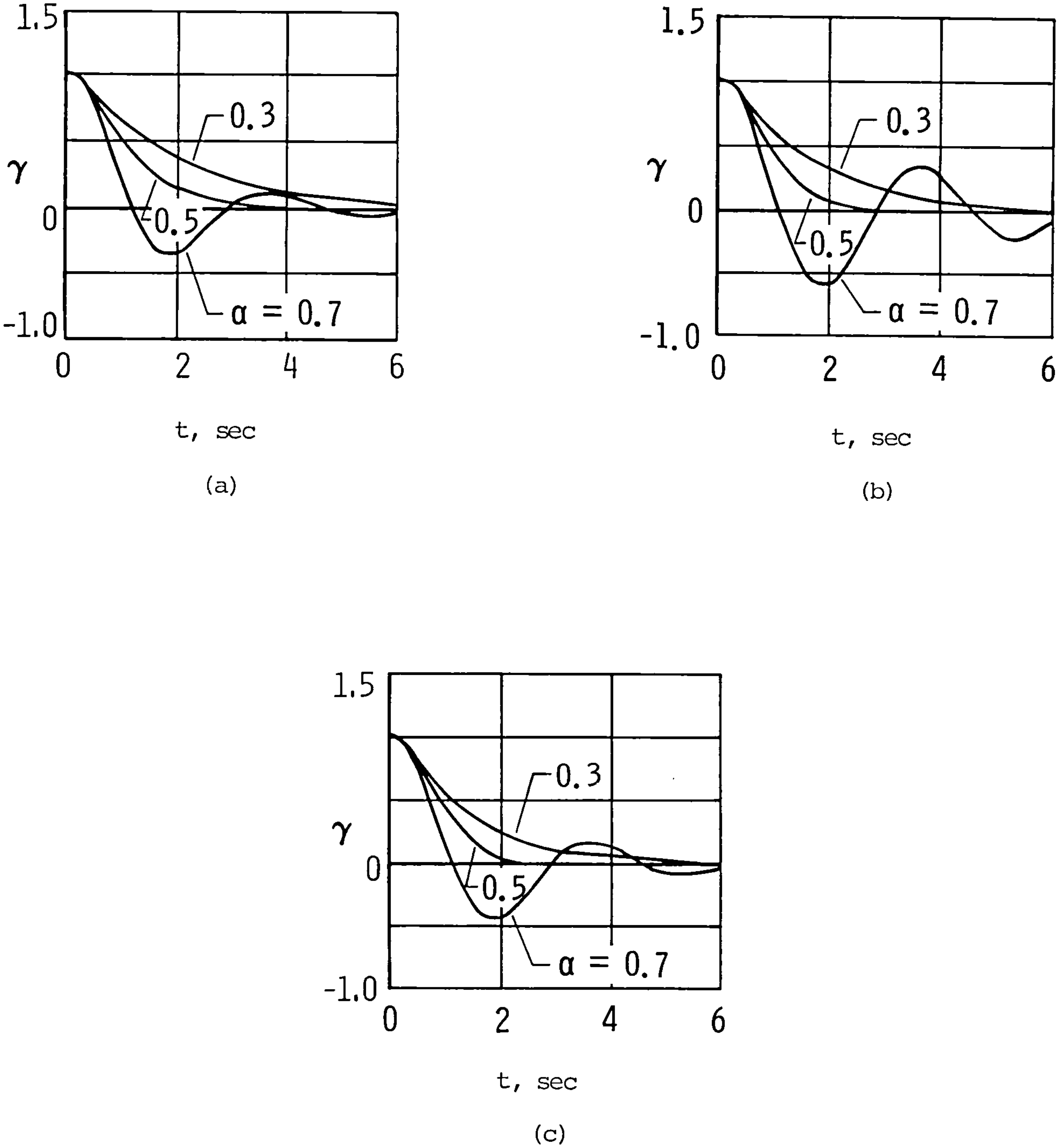

Figure 5. Flight path angle, γ, responses for a) Controller A, b) Controller B, c) Controller C.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-172253		

4. Title and Subtitle

An Application of Nonlinear Porgramming to the Design of Regulators Using a Linear-Quadratic Formulation

5. Report Date
October 1983

6. Performing Organization Code

7. Author(s)

Peter Fleming

8. Performing Organization Report No.
83-60

9. Performing Organization Name and Address
Institute for Computer Applications in Science and Engineering
Mail Stop 132C, NASA Langley Research Center
Hampton, VA 23665

10. Work Unit No.

11. Contract or Grant No.
NAS1-17070

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered
Contractor report

14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor: Robert H. Tolson
Final Report

16. Abstract

A design technique is proposed for linear regulators in which a feedback controller of fixed structure is chosen to minimize an integral quadratic objective function subject to the satisfaction of integral quadratic constraint functions. Application of a nonlinear programming algorithm to this mathematically tractable formulation results in an efficient and useful computer-aided design tool. Particular attention is paid to computational efficiency and various recommendations are made. Two design examples illustrate the flexibility of the apporach and highlight the special insight afforded to the designer.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Nonlinear programming Computer-aided design Control system design Optimal control	31 Engineering (General) Unclassified-Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	26	A03